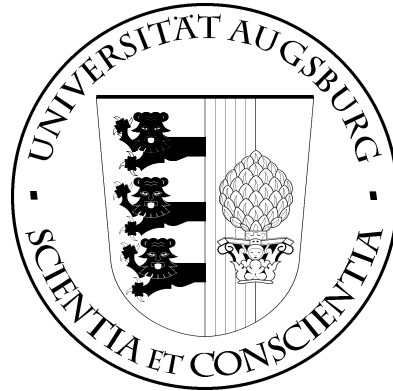


# UNIVERSITÄT AUGSBURG

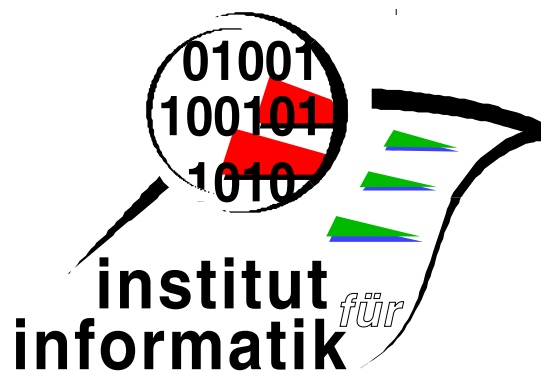


## Learning Scrutable User Models: Inducing Conceptual Descriptions

Martin E. Müller

Report 2002-07

März 2002



INSTITUT FÜR INFORMATIK

D-86135 AUGSBURG



## **Abstract**

The problem of filtering relevant information from the huge amount of available data is tackled by using models of the user's interest in order to discriminate interesting information from un-interesting data. As a consequence, Machine Learning for User Modeling (ML4Um) has become a key technique in recent adaptive systems.

This article presents the novel approach of conceptual user models which are easy to understand and which allow for the system to explain its actions to the user. We show that Ilp can be applied for the task of inducing user models from even sparse feedback by mutual sample enlargement. Results are evaluated independently of domain knowledge within a clear machine learning problem definition.

The whole concept presented is realized in a meta web search engine, OySTER.

(To appear in: "Künstliche Intelligenz", 4/2002.)

# Learning Scrutable User Models: Inducing Conceptual Descriptions

Martin E. Müller

**Abstract.** The problem of filtering relevant information from the huge amount of available data is tackled by using models of the user’s interest in order to discriminate interesting information from uninteresting data. As a consequence, Machine Learning for User Modeling (M4Um) has become a key technique in recent adaptive systems.

This article presents the novel approach of conceptual user models which are easy to understand and which allow for the system to explain its actions to the user. We show that ILP can be applied for the task of inducing user models from even sparse feedback by mutual sample enlargement. Results are evaluated independently of domain knowledge within a clear machine learning problem definition.

The whole concept presented is realized in a meta web search engine, OySTER.

## 1 Conceptual User Models

In M4Um user models are often represented by  $n$ -ary vectors. In the course of content-based document recommendation, the vectors represent significance of “key”-phrases for the user’s interest which is defined as the frequency (TfIdf and similar) of those phrases in the documents the user has rated (‘relevance feedback’) as interesting in the past. In order to decide whether to recommend a document or not, the document (i.e. the corresponding vector) is compared to the user model vector, see [2, 1, 9], [5] and [6].

Though the vectors *represent* a user’s interest, they do not explicitly *describe* a user’s interest: A sequence of word frequencies is not a user model that can be explained to the user in an intuitive way. Thus, our motivation was to find a transparent formalism which is accessible to every user and which allows for an easy translation into the user’s ‘language’. Such a language could be the language of concept hierarchies. The choice of such an approach is supported by the fact, that humans in general (including unexperienced users in the web) are capable of describing and finding objects using concept hierarchies such as library taxonomies, classification systems or web directories. As a result, we obtain a user adaptive system, which is capable of describing internal user models and which can explain recommendations to the user, thereby increasing user acceptance because the user is able to map elements of taxonomies onto meaningful concepts.

The crucial difference is, that the user

## 2 Conceptual descriptions

Within OySTER we use two taxonomies of classes: One for document types ( $\mathcal{T}$ ) and one for document content (categories,  $\mathcal{C}$ ). A pictorial view on the question of whether a certain document is interesting for a user  $u$  with respect to the underlying user model  $\mathbf{M}_u$ , is shown in figure 2.<sup>1</sup>

### 2.1 Document categories and types

The OySTER taxonomy  $\mathcal{C}$  covers research related document in the field of artificial intelligence and neighboring communities. It is handishly coded, taking into account our experience with already existing taxonomies such as the universal decimal classification (Udc), the dmoz ontology and personal experience with classifying news articles, [3]. The resulting ontology consists of 69 classes with a maximum depth of 5.

---

<sup>1</sup> Documents  $a$  and  $b$  are assigned a unique class. Document  $c$  belongs to at least two known classes. Thus, the user is interested in document  $b$  (about machine learning) and not in document  $a$ , because it is about DOS.

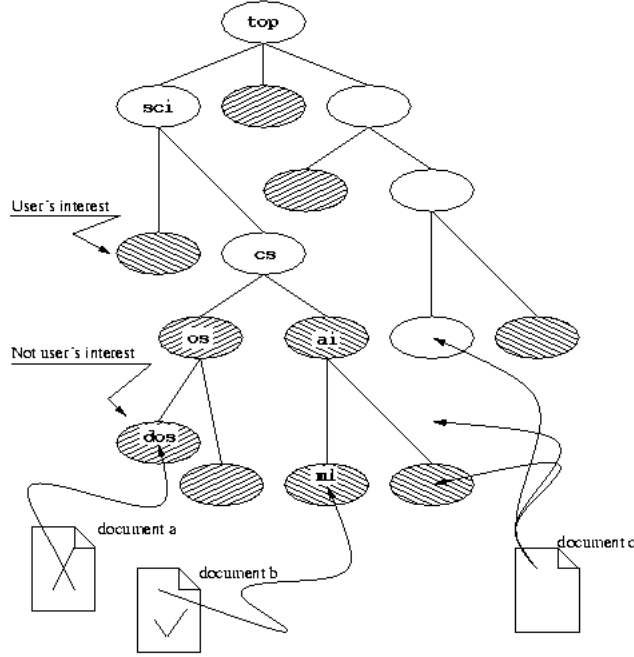


Figure 1: Using  $\mathcal{C}$  for both document classification and user models

Accordingly, the hierarchy of document types can roughly be divided into *homepages*, *research articles* and *lecture material*. Document or text types are discussed within computational linguistics but are rather unknown in the domain of text classification for user-centered document retrieval.<sup>2</sup> The hierarchy of document types contains 35 classes with a maximum tree depth of 4.

The classifiers  $\mathcal{C}$  for the document content taxonomy  $\mathcal{C}$  were developed in course of the Bikini project, [3].

**Representing documents.** Using our classifiers  $\mathcal{C}$  and  $\mathcal{T}$  documents  $d$  are represented by their classification:

$$\begin{aligned} \mathcal{T} \times \mathcal{C}(d) &= \mathcal{T}(d), \mathcal{C}(d) = t : \vec{c} \\ &= t :: p, c_1 :: p_1, c_2 :: p_2, \dots, c_n :: p_n \end{aligned}$$

where  $t$  is the document's type and  $p$  the confidence of the classification and  $c_i$  are categories with decreasing values of confidence  $p_i$ .

Modeling the taxonomies and inheritance of concepts by entailment means that the intuitive subsumption relation *is\_a* is realized as logical implication. Accordingly, the taxonomy is represented as a set of Horn clauses as shown below. The predicate **genpenalty** is used as a generalization bias:

```
cat_..._cs(X,D) :-
    cat_..._cs_programming(X,C),
    genpenalty(C,D).
cat_..._cs_programming(X,D) :-
    cat_..._cs_programming_languages(X,C),
    genpenalty(C,D).
```

<sup>2</sup>Although text types like 'newspaper article', 'obituary' or 'poem' are widely accepted, the 'new' document types that emerged with the Www have not been discussed to a satisfying extent so far. Researchers agree that there are text types like 'homepage' but it is still argued whether those types are already known types which have been adapted to new media or whether they are genuine types.

```
cat_..._cs_programming_languages(X,D) :-
    cat_..._cs_programming_languages_procedural(X,),
    genpenalty(C,D).
```

Document types  $\mathcal{T}$  are represented in a similar way. Background knowledge is represented as a set of facts (here, classification data for the Url 5121):

```
type_..._researchpaper(5121,68).
cat_..._ai_machine_learning_symbolic(5121,92).
cat_..._ai_nat_lang_proc_generation(5121,78).
cat_..._ai_machine_learning_subsymbolic(5121,20).
```

The screenshot shows a web interface for the OySTER search engine. At the top, it says 'Edit into web aspect: Conceptual User Models.' Below this, a note states: 'To delete a conjunct, set its weight to 0. All weights must sum up to 100. No more than 3 conjuncts allowed.' There is a text input field for 'Name:' containing 'Conceptual User Model'. Below this is a table with three rows, each representing a category and its weight. The first row is 'knowledge representation' with a weight of 25. The second row is 'symbolic machine learning' with a weight of 35. The third row is 'user modeling' with a weight of 40. Each row has a checkbox and a weight input field. At the bottom, there is a 'Submit Query' button.

Figure 2: One aspect of a user model as shown in the OySTER search engine

## 2.2 Describing interest by concepts

Since a user is not interested in single, very specific topics only, a user model may consist of several *aspects* which describe different, specific parts of a user’s interest. Aspects can be compared to specific topics of interest, or, e.g. to ‘folders’ for news items or e-mails (c.f. [1, 12]).<sup>3</sup> The aspect shown in figure 2 is interpreted as:

A document  $d$  is considered relevant with respect to the aspect **Conceptual User Models** if it classifies as:

1. a publication about knowledge representation
2. a publication about symbolic machine learning
3. a publication about user modeling

with according confidence values. Representing a user model by a set of Horn clauses is very straightforward. The example from above yields:

$$\begin{aligned}
 \text{p\_interest\_u}(a, D): - \\
 & \text{type\_...publication}(D, T_1), \\
 & \text{cat\_...knowledge\_representation}(D, C_1), \\
 & \text{type\_...publication}(D, T_2), \\
 & \text{cat\_...symbolic\_machine\_learning}(D, C_2), \\
 & \text{type\_...publication}(D, T_3), \\
 & \text{cat\_...user\_modeling}(D, C_3), \\
 & (C_1 \geq 25), (C_2 \geq 35), (C_3 \geq 40),
 \end{aligned} \tag{1}$$

<sup>3</sup>Furthermore, we also explicitly model what the user is *not* interested in (i.e. his *dis-interest*).

### 3 Inducing user models

In the last section we have shown how conceptual user models can be represented as Horn clauses. Our aim is to induce such user models by taking into account user feedback that has been given with respect to documents. Therefore, the learning target is a Prolog clause which gives rise to the application of inductive logic programming methods, Ilp; see [8]. Although Ilp has the advantage of inducing lucid hypotheses, it is still an underestimated machine learning approach in the context of user modeling. Nevertheless, there are several recent advances, e.g. [4]. The hypotheses are generated by the Ilp-learner Progol, [7], which is based on the inverse entailment method.

Our approach and the following evaluation is based on several assumptions and focuses on a special aspect within the whole approach of user adaptive web search:

- (1) Knowledge about documents and knowledge about users are strictly discriminated. We argue, that user models shall contain information about the users interest but no domain knowledge.<sup>4</sup>
- (2) Feedback is strictly discriminated from labels: Feedback (*'relevance feedback'*) is given with respect to documents, but interest is represented by concepts. As a consequence, labels are generated by interpreting feedback as relevance feedback with respect to categories. The way of interpretation allows us to generate small precise or larger but 'noisier' samples
- (3) Following the argument in (1), our approach is based on the assumption, that *no* prior knowledge is available. This means, that our user modeling problem can be described by the aim to learn user models from scratch with only a few examples.

#### 3.1 Representing user models

Thinking towards a working user model induction process and thus having in mind a feasible learning problem, one needs to refine the rough idea of modeling a user's interest a bit further.

One of the most prominent problems within user model induction is sparse negative feedback. This leads to several problems; the most intriguing problem is that one would need to employ a learning algorithm which works for little data and/or for positive data only, [11]. Thus, we introduced the novel approach of explicitly modeling a user's 'disinterest'<sup>5</sup>:

In figure fig:ontosk differently shaded nodes represent concepts that are of interest or which are known not to be of interest. One also could interpret non-interestingness in this figure as explicit disinterest. This leads to the following definition: [User model  $\mathbf{M}_u$ ] A user model  $\mathbf{M}_u$  is a tuple  $M_u^+, M_u^-$ , where  $M_u^+$  models the user's interest and  $M_u^-$  models the user's disinterest. Aspects  $a$  are subsets of  $\mathbf{M}_u$ :  $\mathbf{M}_u(a) = M_u^+(a), M_u^-(a)$ . Interest of  $u$  with respect to aspect  $a$  is represented by the binary predicate  $\mathbf{p\_interest}_u(a, -)$ ; disinterest by the according binary predicate  $\mathbf{n\_interest}_u(a, -)$ . The following clause is an element of  $M_u^+(a) \subseteq M_u^+$ :

$$\begin{aligned} \mathbf{p\_interest}_u(a, D) : - \\ & \mathbf{type\_}t_1(D, T_1), \dots, \mathbf{type\_}t_{n_t}(D, T_{n_t}), \\ & \mathbf{cat\_}c_1(D, C_1), \dots, \mathbf{cat\_}c_{n_c}(D, C_{n_c}), \\ & \mathbf{thresh}(C_1, \vartheta_1), \dots, \mathbf{thresh}(C_n, \vartheta_n). \end{aligned} \tag{2}$$

where  $u$  is the user id,  $a$  the aspect id and  $D$  is instantiated with the id  $d$  of the document currently under consideration. Document types ( $t_i$ ) and categories ( $c_j$ ) are assigned confidence values  $tc_i$  and  $cc_j$ , respectively. Finally, **thresholds** require confidence values to be greater than a certain boundary  $\vartheta_k$ .

Explicitly modeling of the user's disinterest is a rather novel approach in user modeling ([12] use three different kinds of feature vectors to describe long- and short term interest where the short term is described by both explicit interest and disinterest).

<sup>4</sup>Incorporating system dependent domain knowledge will most likely improve the quality of results, but makes evaluation of the learned user model impossible.

<sup>5</sup>We chose this word in analogy to 'dis-like'.

### 3.2 Feedback and samples

In our approach, we assume that feedback  $F_u$  given by a user  $u$  will be used to construct a labeled sample  $\mathbf{f}_u$  using a function  $\Gamma$ . We want to approximate the target function  $\mathbf{i}_u$  (the characteristic function of the user’s interest  $I_u$  on the hypothesis space) by a user model  $\mathbf{M}_u$ .

**Feedback.** The feedback given by user  $u$  is a relation  $F_u$  between documents  $d$  and feedback values.

Relevance of documents with respect to aspect is not necessarily ‘disjoint’, because same document can be interesting to several aspects. For our formalization, we need to assume that feedback is ‘disjoint’. This assumption is required for using feedback given with respect to aspect  $a$  as a basis for larger samples which includes examples for  $a$  as examples for  $a'$  and vice versa. For our evaluation, however, we used feedback data which deliberately violates this assumption (relevance as indicated by feedback may overlap for different aspects) in order to achieve results that withstand real world data.

Accordingly, the sample  $\mathbf{f}_u$  has to provide more information than contained in  $F_u$ . This leads to the following definition: [User modeling sample,  $\mathbf{f}_u$ ] A sample that is used for a learning problem in user modeling consists of a sequence of labeled pieces of evidence

$$\mathbf{f}_u = \left[ d_i, a_i, v_i^{\{+, -\}} \right]_{i \in I} \quad (3)$$

where  $d_i$  are documents of the domain, values  $v_i \in \{+, -\}$  are example labels,  $a_i$  are valid aspects in  $\mathbf{M}_u$  and the superscript denotes the target  $M_u^+$  or  $M_u^-$ . Note, that according to definition 5,  $\mathbf{f}_u$  actually includes label data that will be used for different learning targets, namely  $M_u^+(a_i)$  and  $M_u^-(a_i)$ .

To clarify the contents and information provided by feedback and different samples we have depicted an example in figure fig:feedbackaspect.<sup>6</sup>

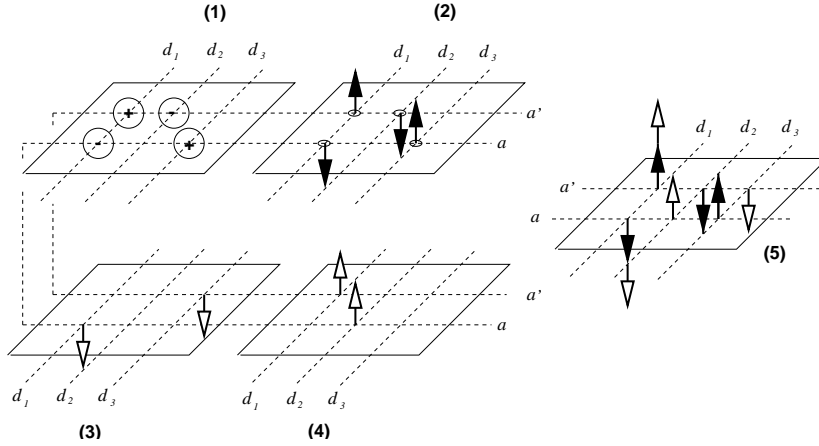


Figure 3: Feedback  $F_{u,a}$  and a sample  $\mathbf{f}_u$

The domain contains three documents  $d_1, d_2, d_3$ . The learning target is split into two aspects  $a, a'$ .

Part (1) shows feedback as given by the user: A plus sign (+) represent a positive label while a minus sign (-) represents negative feedback. Obviously, document  $d_1$  is interesting with respect to  $a'$ , but not with respect to aspect  $a$ . Document  $d_2$  on the other hand, is not interesting with respect to  $a'$ ; its relevance to  $a$  is unknown. The third document is interesting for  $a$ , but relevance

<sup>6</sup> Feedback for document  $d_1$  shows that aspects are disjoint (see end of paragraph def:user-models-dis).



with respect to  $a'$  is unknown. For some documents there is only positive ( $d_3$ ) or only negative evidence ( $d_2$ ).

From this feedback data we want to derive a sample as shown in definition 3.2.

In a first step, (2), the given feedback is mapped one-to-one onto labeled examples. Upward pointing bars indicate feedback with respect to  $M_u^+$ , while downward pointing arrows indicate relevance for disinterest. As an example, consider document  $d_3$ : Since it was rated interesting with respect to  $a$ , we have positive evidence for  $M_u^+(a)$ . But this also means, that  $d_3$  represents negative evidence for  $M_u^-(a')$ . One might claim, that from the feedback gathered we might conclude that  $d_3$  is a negative example for  $M_u^+(a')$  as well: The same holds for  $d_1$  and  $M_u^-(a)$ . This is shown in part (3).

Thinking a step further, one might even guess, that  $d_2$  might be of interest with respect to  $a$ —since it is not of interest with respect to  $a'$ . The same argument applies to  $d_1$  and  $a'$ , see part (4). Of course, this is a very vague guess which may rather cause noisy samples instead of large, reliable samples.

Putting it all together we yield a sample as shown in part (5). Note, that in this figure, we have not depicted whether feedback for some  $d$  with respect to  $a$  is interpreted as evidence for  $M_u^+$  or  $M_u^-$ .

**Generating samples.** As already stated, we assume that a document  $d$  which is interesting for  $u$  with respect to an interest aspect  $a$  is most likely non-interesting for another aspect  $a'$ . Since relevance of  $d$  shall only be provable by means of  $M_u^+(a)$  a hypothesis for  $M_u^+(a')$  should exclude  $d$ . Table fig:generatesamplesfromfeedback shows different ways to interpret relevance feedback using  $\Gamma$ -functions as a sample  $\mathbf{f}_u$  (see last section and figure fig:feedbackaspect). Taking into account

$\Gamma$	$F_{u,a}$		$\mathbf{f}_u$	
	$d, A, v$		$d, a, \mathbf{i}_u(d)$	
	$A$	$v$	$\mathbf{i}_u(d)$	$t$
+	$a$	$\oplus$		$M_u^+$
−	$a$	$\oplus$		$M_u^-$
+	$a$	$\ominus$		$M_u^+$
−	$a$	$\ominus$		$M_u^-$
1	$a'$	$\oplus$		$M_u^+$
2	$a'$	$\oplus$		$M_u^-$
3	$a'$	$\ominus$		$M_u^-$
4	$a'$	$\ominus$		$M_u^+$

Table 1: Generating samples from feedback

relevance feedback with respect to other aspects  $a'$ ,  $\mathbf{f}_u$  can be enlarged by decreasingly reliable information as provided by  $\Gamma_1, \dots, \Gamma_4$ .

$\Gamma_1$  simply states the fact, that aspects are exclusive.  $\Gamma_2$  takes positive evidence for interest in  $a'$  as positive evidence for disinterest in  $a$ . This inference is not as reliable as  $\Gamma_1$ . Therefore,  $\Gamma_2$  should only be considered in cases, where there is very little labeled data available for  $M_u^-$ .

$\Gamma_3$  states that an uninteresting (with respect to  $a'$ ) document  $d$  is not necessarily uninteresting for aspect  $a$ . Thus, it is (weak) negative evidence for  $M_u^-(a)$ , too. However, such documents  $d$  can be interesting for  $M_u^+(a)$ . This is stated by  $\Gamma_4$ , but is not reliable at all and prone to generate noise (thus, it is only defined for reasons of symmetry).

## 4 Results

It is clear, that specific interests provide a better base for inducing user models than shattered or vague feedback.

Accordingly, we carried out an evaluation of several different user characteristics. We simulated four different user characteristics with different specificity of interest. For each user, 50 aspects were generated. By combining aspects we were able to evaluate the learning of single aspects of different specificity as well as the accuracy gain of  $\Gamma$ -application for multiple aspects. The specificity of interest was used to generate more or less noisy feedback samples on simulated Urls with respect to a distance measure on the taxonomy.

The Urls themselves were simulated by generating classification data. To ensure validity of our generated Urls, we defined a diversity measure of Url classifications as the sum of category distances. On real documents, the average diversity was 17.27; the Urls generated in our test set were chosen such that the overall average diversity is slightly worse (18.78).

#### 4.1 Learning single aspects

The left graph in figure 4 shows that accuracy increases both in specificity and sample size.<sup>7</sup> A

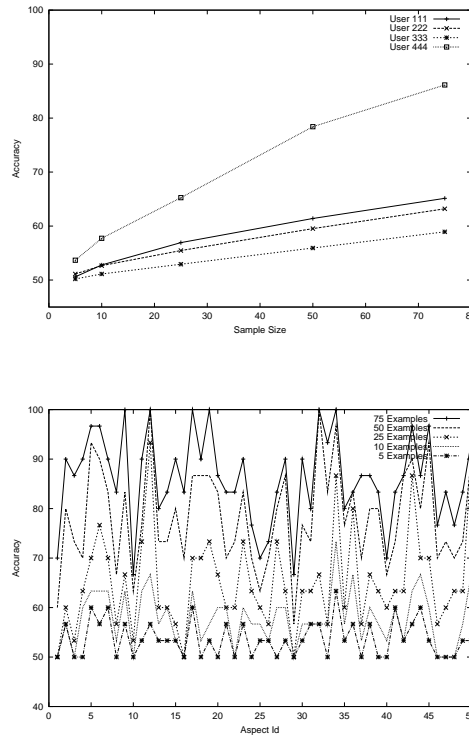


Figure 4: Learning `p_interest` for specific interests

more detailed evaluation of the accuracy for learning `p_interest` for user 444 only is shown in the right graph in figure 4 (the graphs on the left hand side are average values derived from such data files).

#### 4.2 Learning multiple aspects

It has been shown that accuracy for small samples is rather poor; one cannot feel overly impressed by an initial accuracy of 54%. However, learning from five examples only is a very hard problem. This motivates using mutual feedback from other aspects in order to enlarge the sample. In the

<sup>7</sup>Most specific interest in user 444, decreasing specificity for users 111—333.

following evaluation, interest aspect 8 was chosen as aspect  $a'$  whose feedback data is used by  $\Gamma$  in order to gain more examples for learning aspects  $a = [1, \dots, 50]$ .

The evaluation was carried out for users 111–444 with 50 aspects each, thus modeling 200 different aspects of different specificity. For each user, initial samples of length 5, 10, 25, 50 and 75 were generated (with an uniform distribution of positive and negative examples).  $\Gamma_1$  and  $\Gamma_4$  enlarged all samples by 15 examples that were generated from feedback given with respect to aspect  $a' = 8$ . The average accuracy gain in relation to figure 4 is described in table 2. The last line shows results for a worst case evaluation which was performed on deliberately noisy domain data. It shows, that  $\Gamma$  performs best if used on samples of length 25. On shorter samples (length 5 and 10),  $\Gamma$  performs slightly worse (in number and average accuracy gain) than in the reference case. For longer samples, both number of and average accuracy gain are significantly smaller than in the reference case. As a consequence, documents with "multiple topics" (i.e. a strong diversity), generate noisy learning and evaluation samples. A solution to this problem is to take into account only the most reliable document classification. Therefore, the outcome of this evaluation can be regarded as a baseline of minimal performance which is guaranteed by our approach: Accuracy gain is still obtained in several cases and, neglecting cases where  $\Gamma$  actually delivers worse results<sup>8</sup>, follows the general rules of improving learning results.

User	Sample size		
	5	10	50
111	5.8 %	5.6 %	0%
222	5.5 %	6.4 %	0%
333	4.5 %	5.0 %	0%
444	4.4 %	6.6 %	0%
111	3.8 %	3.8 %	2.5 %

Table 2: Average positive accuracy gain

### 4.3 Discussion

From the results, we draw two major conclusions:

**Only small samples should be enlarged.** With growing sample size, the impact of  $\Gamma$  functions decreases. Due to the fact that the accuracy of plainly derived hypotheses increases for larger samples, application of  $\Gamma$  functions does not contribute to the result. In such cases, enlarged samples noisify the input. As results for sample length 50 show, accuracy on  $\Gamma$  samples actually drops significantly below plain learning accuracy.

**Specificity of interest.** Over-specific interest as simulated by user 444 leads to highly accurate hypotheses. This explains a relatively poor accuracy gain using  $\Gamma$  functions for user 444. The best average accuracy gain (5.8%) on samples of length 5 was obtained for the specific (but not over-specific) user 111. More than 20% of all aspects could be learned with a higher accuracy using  $\Gamma$ -functions. For a sample size of 10 examples, similar results can be derived. It is noteworthy, that  $\Gamma$  helps to increase accuracy for 9 of 50 aspects for user 111 while it helps for only 3 aspects for user 444.

It has to be emphasized, that we deliberately violated our assumption of ‘disjointness’ of aspects in this evaluation (the extreme case is the result for learning aspect 8) in order to show that our approach even works in domains which do not meet the theoretical requirements. A better method to cope with similar aspects would have been to choose only those aspects  $a', a'', \dots$

<sup>8</sup>Such cases can be prevented by testing sample accuracy of hypothesis for plain samples and  $\Gamma$ -enhanced samples. If sample accuracy for enhanced samples drops below accuracy on the initial sample, hypotheses are discarded.

for each learning task  $M_u(a)$ , where  $a', a'', \dots$  significantly differ from  $a$  (this can be modeled by means of the distance measure  $\delta$  on  $\mathcal{C}$ ).

**Summary of results.** It has been shown, that sample enlargement pays off for small samples only.

Plain learning average accuracy results of between 50 to 52% do not sound very promising.

One must bear in mind though, that samples of length 5 pose a very hard learning problem. When taking this into account one can conclude that an accuracy gain of 10% (in relation to accuracy levels by plain learning) can be yielded in up to 20% of all cases. Especially in cases where there is very little feedback available (namely 5 or 10 examples) an accuracy gain of 5% should be viewed as a major improvement.

## 5 Conclusion

In this article, we have shown, that user models representing a user's interest can be represented as conceptual descriptions. Conceptual user models can be translated into Prolog clauses such that the problem of learning user models becomes a problem of inducing logic programs. We presented a method for coping with sparse feedback and small samples.

Together, this forms the novel idea of inducing conceptual user models using the rather uncommon method of ILP. One major advantage of the approach is that scrutability and explainability are guaranteed as a side effect of the chosen method: User models are sets of Prolog clauses which can be easily explained to the user because each body literal corresponds to a class of the taxonomy  $\mathcal{C}$ . Furthermore, recommendation of results to the user can be explained as well: (dis-)interestingness of a document is determined by carrying out Prolog proofs. Results show relevance of documents with respect to aspects. As a side-effect, results are disambiguated: A query for "compression" yields results which are evaluated with respect to classes "information theory", "file systems" and "diving" and according aspects. The proofs themselves can be verbalized as well, also taking into account inheritance between classes of the taxonomy.

Since the problem of learning a user model can be properly dissected from its surroundings, we evaluated the approach without veiling the facts by way of additional background knowledge. As a consequence, the results obtained can easily be outstripped by approaches which either take into account more information (domain knowledge word occurrences) or impose severe restrictions on the domain (like independence of features).

Nevertheless, there remain many open research questions. In this article, we did not explain the underlying (assymetric) distance measure we defined on the taxonomy. It remains to be evaluated, how different distance measure affect the learning results. Another interesting research question is, how so-called redundant hypotheses could be used to infer more accurate hypotheses: During the induction process many hypotheses are generated (consisting of up to 15 literals) which only cover single examples. Regarding compression measures for hypotheses, those clauses are discarded. Still, they carry valuable information: One could imagine, that we collect all literals from all redundant clauses and then construct a new clause using an information gain driven method (see Foil, [10]).

## References

- [1] BALABANOVIC, M. An interface for learning multi-topic user profiles from implicit feedback. In *AAAI-98 Workshop on Recommender Systems* (Madison, Wisconsin, July 1998).
- [2] BILLSUS, D., AND PAZZANI, M. A hybrid user model for news story classification. Springer, 1999, pp. 99–108.

- [3] EILERT, S., MENTRUP, A., MÜLLER, M. E., ROLF, R., ROLLINGER, C.-R., SIEVERTSEN, F., AND TRENKAMP, F. Bikini: User adaptive news classification in the world wide web. In *Workshop "Machine Learning for User Modeling", 8th Intl. Conf. on User Modeling* (2001).
- [4] JACOBS, N., AND BLOCKEEL, H. The learning shell: Automated macro construction. In *Proc. 8th Intl. Conf. User Modeling* (2001), M. Bauer, P. Gmytrasiewicz, and J. Vassileva, Eds., Springer.
- [5] JOACHIMS, T., FREITAG, D., AND MITCHELL, T. Webwatcher: A tour guide for the world wide web. In *Proceedings of IJCAI 97* (August 1997).
- [6] LIEBERMAN, H. Letizia: An agent that assists web browsing. In *Proc. of the IJCAI 95* (1995), pp. 924–929.
- [7] MUGGLETON, S. Inverse entailment and Progol. *New Generation Computing*, 13 (1995), 245–286.
- [8] MUGGLETON, S., AND RAEDT, L. D. Inductive logic programming: Theory and methods. *Machine Learning* 19, 20 (1994), 629–679.
- [9] PAZZANI, M., MURAMATSU, J., AND BILLSUS, D. Syskill & Webert: Identifying interesting Web Sites. In *Proc. of the National Conference on Artificial Intelligence* (Portland, OR, 1996).
- [10] QUINLAN, J. R. Learning logical definitions from relations. *Machine Learning* 5, 3 (1990), 239 – 266.
- [11] SCHWAB, I., KOYCHEV, I., AND KOBSA, A. Learning about users from observation. In *Adaptive User Interfaces* (Menlo Park, California, March 2000), S. Rogers and W. Iba, Eds., no. SS-00-01 in Technical Report, AAAI Press, pp. 102–106.
- [12] WIDYANTORO, D. H., YIN, J., NASR, M. S. E., YANG, L., ZACCHI, A., AND YEN, J. Alipes: A swift messenger in cyberspace. In *Proc. AAAI Spring Symposium on Intelligent Agents in Cyberspace* (Palo Alto, March 1999).